

Deciding Entailment of Implications with Support and Confidence in Polynomial Space

Daniel Borchmann

TU Dresden

Faculty of Mathematics and Sciences

Institute for Algebra

`daniel.borchmann@mailbox.tu-dresden.de`

January 31, 2012

Abstract

Association Rules are a basic concept of data mining. They are, however, not understood as logical objects which can be used for reasoning. The purpose of this paper is to investigate a model based semantic for implications with certain constraints on their support and confidence in relational data, which then resemble association rules, and to present a possibility to decide entailment for them.

1 Introduction

Association Rules are a basic pattern in data mining which has undergone extensive research during the last decades. Thereby a major focus has been on the question how to represent the set of all association rules as concisely as possible, as this set might be too large to be practically helpful. However, in those considerations representations based on model semantics have not been used. Instead, rules based formalisms for inferring association rules from others have been developed [7, 5]. One notable exception is the work of Luxemburger [6] who investigated *partial implication*, which are closely related to association rules and to the constrained implications we shall introduce in this paper.

In this work we want to investigate this model based approach. For this, we shall develop a straight forward model based semantic for *constrained implications*, which shall replace the original notion of association rules. The main focus of our work then shall be the investigation on whether a set of constrained implications *entails* another constrained implication or not. We shall show that this question can be rephrased in terms of two linear programs, which then can be solve in polynomial space.

We shall start our investigation with a short introduction into Formal Concept Analysis [1], which we are going to use as the basic formalism to formulate our considerations. After this, we introduce the classical concepts of *support* and *confidence*. We

then investigate constrained implications and formulate a semantic based on models for constrained implications. We shall finally formulate the entailment problem for constrained implications.

The next step shall then be an equivalent reformulation of a certain instance of the entailment problem into a set of two linear programs. The reformulation we are going to make is quite natural, however the resulting pair of linear programs may be exponentially large in the size of the original input. Nevertheless, as we shall show subsequently, we are able to show that we can solve both linear programs in space polynomial in the size of the input. Hence, entailment of constrained implications is decidable in PSpace.

2 Implications with Support and Confidence

Let G and M be two sets and let $I \subseteq G \times M$. We shall call the triple $\mathbb{K} = (G, M, I)$ a *(formal) context* and shall associate with it the following interpretation: The elements $g \in G$ are the *objects* of \mathbb{K} , the elements $m \in M$ are the *attributes* of \mathbb{K} and we shall say that g *has the attributes* m if and only if $(g, m) \in I$. In that case we may also write $g \text{ } I \text{ } m$. The set of objects of a formal context \mathbb{K} may be denoted $G_{\mathbb{K}}$, likewise $M_{\mathbb{K}}$ denotes the set of attributes of \mathbb{K} and $I_{\mathbb{K}}$ denotes the incidence relation of \mathbb{K} .

Let us fix a formal context \mathbb{K} and let $A \subseteq M_{\mathbb{K}}$. We shall denote with

$$A' := \{ g \in G \mid \forall m \in A : g \text{ } I \text{ } m \}$$

the set of all common objects of all attributes of A . Likewise, for a set $B \subseteq G(\mathbb{K})$ the set of common attributes of all objects in B is denoted by

$$B' := \{ m \in M \mid \forall g \in B : g \text{ } I \text{ } m \}.$$

The sets A' and B' are called the *derivations* of A and B in \mathbb{K} , respectively, and accordingly the operators \cdot' are the called *derivation operators* of \mathbb{K} .

Let $A, B \subseteq M_{\mathbb{K}}$. We shall call the pair (A, B) an *implication* of \mathbb{K} and denote it with $A \longrightarrow B$. The set of all implications of \mathbb{K} is denoted by $\text{Imp}(\mathbb{K})$. We shall say that the implication $A \longrightarrow B$ *holds* in \mathbb{K} , denoted $\mathbb{K} \models A \longrightarrow B$, if and only if every object $g \in G_{\mathbb{K}}$ that has all the attributes of A also has all the attributes of B . Using the derivation operators, we can comprehensively write this as

$$\mathbb{K} \models A \longrightarrow B \iff A' \subseteq B'.$$

Now let \mathbb{K} be a finite and non-empty formal context, i. e. both $G_{\mathbb{K}}$ and $M_{\mathbb{K}}$ are now assumed to be finite and non-empty sets. The *support of* A is defined to be

$$\text{supp}_{\mathbb{K}}(A) := \frac{|A'|}{|G|}.$$

Similarly, the *support of the implication* $A \longrightarrow B$ is set to be

$$\text{supp}_{\mathbb{K}}(A \longrightarrow B) := \text{supp}_{\mathbb{K}}(A).$$

Furthermore, the *confidence* of the implication $A \longrightarrow B$ is given by

$$\text{conf}_{\mathbb{K}}(A \longrightarrow B) := \frac{|(A \cup B)'|}{|A'|}$$

if $|A'| \neq 0$. Otherwise, $\text{conf}_{\mathbb{K}}(A \longrightarrow B) := 1$.

We are now ready to introduce the notion of constrained implications.

Definition 1 Let \mathbb{K} be a finite and non-empty formal context. Then a *constrained implication* of \mathbb{K} is a triple $(A \longrightarrow B, s, c)$ where $A \longrightarrow B \in \text{Imp}(\mathbb{K})$ and $s, c \in [0, 1] \cap \mathbb{Q}$.

A constrained implication $r = (A \longrightarrow B, s, c)$ is said to have *minimal support* s_0 and *minimal confidence* c_0 if $s \geq s_0$ and $c \geq c_0$. We shall denote the set of all constrained implications of \mathbb{K} with minimal support s_0 and minimal confidence c_0 by $\text{conImp}_{s,c}(\mathbb{K})$. \diamond

Constrained implications resemble the notion of association rules, but in a more formal setting. In particular, we can ask whether a constrained implication *holds* in an *arbitrary* formal context or not.

Definition 2 Let \mathbb{K} be a finite and non-empty formal context and let $r = (A \longrightarrow B, s, c)$ be constrained implication of \mathbb{K} . Then r *holds* in \mathbb{K} , written as $\mathbb{K} \models r$, if and only if

$$\begin{aligned} \text{supp}_{\mathbb{K}}(A \longrightarrow B) &\geq s, \\ \text{conf}_{\mathbb{K}}(A \longrightarrow B) &\geq c. \end{aligned} \quad \diamond$$

Let us fix a set $\mathcal{L} = \{(A_i \longrightarrow B_i, s_i, c_i) \mid i \in I\}$ of constrained implications and let $r = (A \longrightarrow B, s, c)$ be another constrained implication. The problem we want to investigate in the sequel is the following.

Definition 3 (Entailment for Constrained Implications)

Given \mathcal{L} and $(A \longrightarrow B, s, c)$, does then for every formal context \mathbb{K} hold that

$$\mathbb{K} \models \mathcal{L} \implies \mathbb{K} \models (A \longrightarrow B, s, c),$$

i. e. does \mathcal{L} *entail* $(A \longrightarrow B, s, c)$? \diamond

We shall write $\mathcal{L} \models (A \longrightarrow B, s, c)$ if and only if \mathcal{L} entails $(A \longrightarrow B, s, c)$. An *instance* of the entailment problem is just a pair (\mathcal{L}, r) of a set of constrained implications \mathcal{L} and a single constrained implication r .

3 Linear Programs for the Entailment Problem

The purpose of this section is to association with the set \mathcal{L} and the rule $r = (A \longrightarrow B, s, c)$ a pair of linear programs such that entailment can be decided by solving the

programs. Before we do so, however, we shall somehow restrict the sets of possible formal contexts \mathbb{K} that we have to investigate to decide entailment.

For this we observe that we can restrict the set of attributes of the formal contexts occurring in the entailment problem to those attributes which actually occur somewhere in the implications involved. Define

$$M := (A \cup B) \cup \bigcup_{i \in I} A_i \cup B_i.$$

Then the following statement holds:

Lemma 4 *Let \mathcal{L} and $r = (A \longrightarrow B, s, c)$ as before. Then $\mathcal{L} \models (A \longrightarrow B, s, c)$ if and only if for all formal contexts \mathbb{K} with attribute set M it holds that*

$$\mathbb{K} \models \mathcal{L} \implies \mathbb{K} \models (A \longrightarrow B, s, c).$$

Proof The direction (\implies) is clear. The main idea for the other direction is that implications can only “restrict” the occurrence of attributes they contain. More formally, let us assume that $\mathcal{L} \not\models r$. We shall show that then there exists a finite and non-empty formal context \mathbb{K} with attributes set M such that $\mathbb{K} \models \mathcal{L}$ and $\mathbb{K} \not\models r$. Since $\mathcal{L} \not\models r$ there exists a formal context \mathbb{K}_0 with $M \subseteq M_{\mathbb{K}_0}$ such that $\mathbb{K}_0 \models \mathcal{L}$ and $\mathbb{K}_0 \not\models r$. Let \mathbb{K} be the formal context that arises from \mathbb{K}_0 by deleting all attributes $M_{\mathbb{K}_0} \setminus M$. More formally,

$$\mathbb{K} := (G_{\mathbb{K}_0}, M, I_{\mathbb{K}_0} \cap (G_{\mathbb{K}_0} \times M)).$$

Now if $X \subseteq M$, then the derivation of X in \mathbb{K}_0 and \mathbb{K} is the same, since

$$g I_{\mathbb{K}_0} m \iff g I_{\mathbb{K}} m$$

holds for all $g \in G_{\mathbb{K}_0}$ and $m \in X$. Therefore, the support and confidence of all elements in \mathcal{L} and of the rule r is the same on both \mathbb{K}_0 and \mathbb{K} and hence $\mathbb{K} \models \mathcal{L}, \mathbb{K} \not\models r$ as required. \square

We are now going to construct the aforementioned linear programs for our instance (\mathcal{L}, r) of the entailment problem.

Let \mathbb{K} be a formal context with attribute set M . For every set $A \subseteq M$ let x_A be the number of objects g in \mathbb{K} such that $g' = A$, divided by $|G|$, i. e.

$$x_A := \frac{|\{g \in G \mid g' = A\}|}{|G|}.$$

Now if $r_i = (A_i \longrightarrow B_i, s_i, c_i) \in \mathcal{L}$ is such that $\mathbb{K} \models r_i$, then

$$\begin{aligned} \text{supp}_{\mathbb{K}}(A_i \longrightarrow B_i) &= \text{supp}_{\mathbb{K}}(A_i) \geq s_i, \\ \text{conf}_{\mathbb{K}}(A_i \longrightarrow B_i) &\geq c_i. \end{aligned}$$

This can be rewritten with the help of the variables x_A as

$$\begin{aligned} \sum_{A \supseteq A_i} x_A &\geq s_i, \\ \sum_{A \supseteq A_i \cup B_i} x_A - c_i \sum_{A \supseteq A_i} x_A &\geq 0. \end{aligned} \tag{1}$$

The second inequality stems from the fact that $\text{supp}_{\mathbb{K}}(A_i \cup B_i) - c_i \text{supp}_{\mathbb{K}}(A_i) \geq 0$.

Additionally, the variables x_A satisfy the equation $\sum_{A \subseteq M} x_A = 1$, which can be rewritten as two inequalities:

$$\begin{aligned} \sum_{A \subseteq M} x_A &\geq 1, \\ - \sum_{A \subseteq M} x_A &\geq -1. \end{aligned} \tag{2}$$

Finally, all values x_A satisfy $x_A \geq 0$ and $x_A \in \mathbb{Q}$.

Those inequalities can be rewritten in a compact form using matrices. Let $\mathbf{A} \in \mathbb{Q}^{(2|I|+2) \times 2^{|M|}}$ and $\mathbf{b} \in \mathbb{Q}^{2|I|+2}$ be such that the aforementioned inequalities are represented as

$$\mathbf{A}\mathbf{x} \geq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}. \tag{3}$$

Thereby the vector $\mathbf{x} = (x_i)_{i=1, \dots, 2^{|M|}}$ shall be such that the variable x_i corresponds to the variable x_A where the binary representation of i corresponds to the characteristic vector of A as a subset of M , for some arbitrary but fixed linear ordering of M .

Example 5 Let us illustrate the construction by means of a simple example. Let

$$\mathcal{L} = \{ (\{a\} \longrightarrow \{b\}, 1/2, 1/3), (\{a\} \longrightarrow \{c\}, 1/3, 1/4) \}.$$

Then $M = \{a, b, c\}$. Let us order M by virtue of $a > b > c$. Then

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1/3 & -1/3 & 1 - 1/3 & 1 - 1/3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & -1/4 & 0 & -1/4 & 0 & 1 - 1/4 & 0 & 1 - 1/4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 1/2 \\ 0 \\ 1/3 \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

Here we have used the aforementioned convention, namely that the index i corresponds to the subsets A_i of M whose characteristic vector with respect to the chosen linear ordering of M is the binary representation of i . Hence, $i = 0$ corresponds to the empty set, $i = 1$ to the set $\{c\}$ and so on. \diamond

Theorem 6 *Let $\mathcal{L} = \{(A_i \longrightarrow B_i, s_i, c_i) \mid i \in I\}$ be a set of constrained implications and let $M := \bigcup_{i \in I} A_i \cup B_i$. Then every context \mathbb{K} with attribute set M and $\mathbb{K} \models \mathcal{L}$ induces a solution $\mathbf{x}_{\mathbb{K}}$ of (3). Conversely, every solution \mathbf{x} of (3) induces a nonempty collection \mathcal{K} of formal context \mathbb{K} with attribute set M , $\mathbb{K} \models \mathcal{L}$ and $\mathbf{x}_{\mathbb{K}} = \mathbf{x}$.*

We therefore see that there exists a one-to-one correspondence between non-empty classes of models of \mathcal{L} and the solutions of (3).

Proof From the previous considerations we have seen that every formal context \mathbb{K} with $\mathbb{K} \models \mathcal{L}$ gives rise to a solution of (3).

For the converse direction let \mathbf{x} be a solution of (3). Then the entries x_i of \mathbf{x} are rational numbers. Let n be the least common denominator of all entries of x_i . Then define the formal context $\mathbb{K}_{\mathbf{x}}$ with attribute set M as follows: For every index $i \in \{1, \dots, 2^{|M|}\}$ we add $x_i \cdot n$ objects to $\mathbb{K}_{\mathbf{x}}$ whose intent is the set A that corresponds to the binary representation of i . Then $|G_{\mathbb{K}_{\mathbf{x}}}| = n$ and we claim that $\mathbb{K}_{\mathbf{x}}$ is a model of \mathcal{L} .

For this let $i \in I$. Then

$$\text{supp}_{\mathbb{K}_{\mathbf{x}}}(A_i \longrightarrow B_i) = \sum_{j \mid A_j \supseteq A_i} x_j \geq s_i$$

where A_j is the corresponding set for the index j . Likewise it follows $\text{conf}_{\mathbb{K}_{\mathbf{x}}}(A_i \longrightarrow B_i) \geq c_i$, because \mathbf{x} is a solution for (3). From the construction of $\mathbb{K}_{\mathbf{x}}$ and the construction of $\mathbf{x}_{\mathbb{K}_{\mathbf{x}}}$ it is apparent that $\mathbf{x} = \mathbf{x}_{\mathbb{K}_{\mathbf{x}}}$. \square

The crucial observation now is that we can transform the problem of deciding entailment into a pair of linear optimization problems. The key idea is that we can read the entailment problem $\mathcal{L} \models (A \longrightarrow B, s, c)$ as the question whether the minimal possible support for $A \longrightarrow B$ in every model of \mathcal{L} can be lower than s ; similarly, one can ask what the minimal confidence of $A \longrightarrow B$ in all models of \mathcal{L} is lower than c . Therefore, entailment is equivalent to solving two linear optimization problems over \mathbb{Q} .

Theorem 7 *Let $\mathcal{L} \cup \{(A \longrightarrow B, s, c)\}$ be a set of constrained implications, $|M| = \bigcup_{i \in I} A_i \cup B_i \cup A \cup B$ and let \mathbf{A} and \mathbf{b} as defined for (3). Furthermore, let*

$$\mathcal{X} := \{\mathbf{x} \in \mathbb{Q}^{2^{|M|}} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Then $\mathcal{L} \models (A \longrightarrow B, s, c)$ if and only if

$$\begin{aligned} s &\leq \min\left\{\sum_{X \supseteq A} x_X \mid \mathbf{x} \in \mathcal{X}\right\} \\ 0 &\leq \min\left\{\sum_{X \supseteq A \cup B} x_X - c \sum_{X \supseteq A} x_X \mid \mathbf{x} \in \mathcal{X}\right\}. \end{aligned} \tag{4}$$

Proof Let $\mathcal{L} \models (A \longrightarrow B, s, c)$. Then for every formal context \mathbb{K} with attribute set M and $\mathbb{K} \models \mathcal{L}$ it holds

$$\begin{aligned} \text{supp}_{\mathbb{K}}(A \longrightarrow B) &\geq s, \\ \text{conf}_{\mathbb{K}}(A \longrightarrow B) &\geq c, \end{aligned}$$

and so for the corresponding solution $\mathbf{x}_{\mathbb{K}}$ of (3) we have

$$\begin{aligned} \sum_{X \supseteq A} x_X &\geq s, \\ \sum_{X \supseteq A \cup B} x_X - c \sum_{X \supseteq A} x_X &\geq c. \end{aligned}$$

Since by Theorem 6 every solution of (3) originates from a model \mathbb{K} of \mathcal{L} , the inequalities (4) are satisfied.

Conversely, let $\mathcal{L} \not\models (A \longrightarrow B, s, c)$. Then by Lemma 4 there exists a formal context \mathbb{K} with attribute set M such that $\mathbb{K} \models \mathcal{L}$ and $\mathbb{K} \not\models (A \longrightarrow B, s, c)$, i. e.

$$\begin{aligned} \text{supp}_{\mathbb{K}}(A \longrightarrow B) &< s \text{ or} \\ \text{conf}_{\mathbb{K}}(A \longrightarrow B) &< c. \end{aligned}$$

Then the corresponding solution $\mathbf{x}_{\mathbb{K}}$ of (3) satisfies

$$\begin{aligned} \sum_{X \supseteq A} x_X &< s \text{ or} \\ \sum_{X \supseteq A \cup B} x_X - c \sum_{X \supseteq A} x_X &< c. \end{aligned}$$

and so the inequalities (4) are not satisfied. \square

4 Deciding Entailment in Polynomial Space

The main disadvantage of the above approach is the size of the system (3), which is exponential in $|M|$ and might therefore be exponential in the size of the original input \mathcal{L} and $r = (A \longrightarrow B, s, c)$. The resulting linear programs (4) might therefore have a number of variables which is exponential in the number of inequalities. The case, however, that the number of variables is much larger than the number of restrictions is quite common in linear programming and algorithms have been devised to handle them specially.

The purpose of this section is to represent such an algorithm. We shall show that this algorithm then is able to solve (4) in space polynomial in the size of the original input \mathcal{L} and r .

To properly formulate this claim we need to agree on an encoding system for \mathcal{L} and r , for otherwise the notion of *polynomial space in the size of the input* would be meaningless. See [2] for more details.

We shall agree on the convention that integers are stored as binary numbers. Rational numbers can be stored as pairs of integers. Matrices and vectors are stored as sequences of their entries, in row-major order. Finally, sets are represented as sequences of the elements they contain. For convenience we shall assume that we can use special characters like braces, parentheses and commas to separate different entities. This only increases the space requirements by a negligible amount.

As the following description of the algorithm gets technical at certain points we may adopt some syntactical conventions to make this description more pragmatic.

If \mathbf{x} is a rational vector, we may write $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ to denote two vectors $\mathbf{x}_1, \mathbf{x}_2$ whose entries are a partition of the entries of \mathbf{x} . That does not necessarily mean that the vector \mathbf{x} must be ordered such that \mathbf{x}_1 comes first and \mathbf{x}_2 second. The same notation may be used to denote column partitions of matrices, i.e. that the columns of two matrices $\mathbf{A}_1, \mathbf{A}_2$ constitute the columns of a matrix \mathbf{A} may be written as $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2)$, without any special assumptions on the ordering of any columns. Finally, the columns of a matrix \mathbf{A} are denoted by $(\mathbf{a}_i)_{i=1, \dots, n}$, if \mathbf{A} has n columns.

4.1 The Revised Simplex Method

One of the best known algorithms for solving linear programs is the *Simplex Method* devised by George Dantzig. The *revised Simplex Method* is a variation of the original Simplex Method that reduces the space needed for the actual computations. The revised Simplex Method is mathematical folklore and we shall only give a short and compact representation of the method without giving any proofs. For further details, we refer to any introductory text on linear programming. The following description follows [4, 3].

Let us consider the general linear optimization problem

$$\max\{\mathbf{c}^T \mathbf{x} \mid \mathbf{x} \in \mathbb{Q}^n, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \quad (5)$$

where $\mathbf{A} \in \mathbb{Q}^{m \times n}$ and $\mathbf{b} \in \mathbb{Q}^m$. Moreover, we assume $m \leq n$ and that \mathbf{A} has maximal rank, i.e. $\text{rank}(\mathbf{A}) = m$. Then the vertexes of the polytope¹ $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}$ contain optimal solutions for (5), and the Simplex Method (as well as the revised Simplex Method) goes along the edges of the polytope to find an optimal vertex. Vertexes of P correspond to maximal independent sets $B = \{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_k}\}$ of certain columns of \mathbf{A} , and since \mathbf{A} has full rank, $k = m$ and the set B can be understood as a regular submatrix \mathbf{B} of \mathbf{A} . Let us denote with \mathbf{N} the submatrix of \mathbf{A} that contains all columns except those in B .

Starting with such a submatrix \mathbf{B} , which corresponds to a vertex \mathbf{x} of P , the revised Simplex Method now tries to find another vertex \mathbf{x}' such that $\mathbf{c}^T \mathbf{x} \leq \mathbf{c}^T \mathbf{x}'$. If such a vertex exists, then there also exists a vertex which is linked to \mathbf{x} by an edge

¹A *polytope* is finite intersection of halfspaces in \mathbb{Q}^n .

of the polytope P . To find such a vertex \mathbf{x}' , which amounts to find a set B' or the corresponding submatrix \mathbf{B}' of \mathbf{A} , the revised Simplex Method proceeds as follows:

- i. Subdivide the vectors $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$ and $\mathbf{c} = (\mathbf{c}_B, \mathbf{c}_N)$ according to the chosen columns of \mathbf{A} in B . Then $\mathbf{x}_N = \mathbf{0}$ and $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$.
- ii. Compute $\mathbf{r} = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$.
- iii. If all entries in \mathbf{r} are non-positive, then \mathbf{x} is an optimal solution for (5).
- iv. Otherwise let k be such that the k th entry in \mathbf{r} is negative.
- v. If all entries in the k th column of $\mathbf{B}^{-1} \mathbf{N}$ are non-positive, then (5) does not have a solution (the value of $\mathbf{c}^T \mathbf{x}$ is unbounded over P). Abort.
- vi. Otherwise let s be an index such that $\frac{x_s}{d_{sk}}$ is minimal with $d_{sk} > 0$. Here x_j is the j th component of \mathbf{x}_B and the entries of $\mathbf{B}^{-1} \mathbf{N}$ are the elements d_{ij} . Thereby choose s such that the corresponding column \mathbf{a}_s of \mathbf{A} is the lexicographically smallest column such that $\frac{x_s}{d_{sk}}$ is minimal.
- vii. Now the s th column \mathbf{a}_s of \mathbf{A} is an element of B and the k th column \mathbf{a}_k of \mathbf{A} is not an element of B . Define $B' := B \setminus \{\mathbf{a}_s\} \cup \{\mathbf{a}_k\}$.

It can be shown that the solution \mathbf{x}' corresponding to the set B' is a vertex of P such that $\mathbf{c}^T \mathbf{x}' \geq \mathbf{c}^T \mathbf{x}$. Iterating this procedure until it either returns an optimal solution or the information that no optimal solution exists constitutes the optimization procedure of the revised Simplex Algorithm. By (10.6) of [4], this algorithm always terminates.

Now that we know the necessary details of the revised Simplex Method, we are able to prove the following theorem.

Theorem 8 *Let $\mathbf{A} \in \mathbb{Q}^{m \times n}$, $\mathbf{b} \in \mathbb{Q}^m$, $\mathbf{c}, \mathbf{x}' \in \mathbb{Q}^n$ such that $m \leq n$, $\text{rank}(\mathbf{A}) = m$ and $\mathbf{A}\mathbf{x}' = \mathbf{b}$. Let k be the largest amount of space needed to store any entry of $\mathbf{A}, \mathbf{b}, \mathbf{c}$. Then the linear program*

$$\max \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{x} \in \mathbb{Q}^n, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}$$

can be solved in space $\mathcal{O}(km^3)$ in addition to the space needed to represent the matrix \mathbf{A} and the vectors \mathbf{b} and \mathbf{c} .

Proof To see that the statement holds we have to show that we can execute all steps of the revised Simplex Method in not more than $\mathcal{O}(km^3)$ space or that we can replace them by equivalent computations which can be done in space $\mathcal{O}(km^3)$.

We start by observing that we only need to store the indexes of the elements of B , \mathbf{x}_B and \mathbf{c}_B . This can be done in space $\mathcal{O}(km)$. Computing the inverse can be done in space $\mathcal{O}(km^3)$, using the adjugate matrix of \mathbf{B} . This involves computations of determinants of \mathbf{B} and of submatrices of \mathbf{B} with one less row and column. Those computations can be carried out in time $\mathcal{O}(km^3)$ each and hence do not need more space than $\mathcal{O}(km^3)$.

The steps (ii)–(iv) can be replaced by a simple loop that computes every component of \mathbf{r} separately and returns the index of the first positive component. For this we note that we only need to compute $\mathbf{c}_B^T \mathbf{B}^{-1}$ once, yielding a row vector. After this, we successively compute scalar products with rows of \mathbf{N} and save the index of the first row where the result is negative. If there is no such index, the algorithm stops. All this can be done in space $\mathcal{O}(km)$.

It is clear that we can execute step (v) in linear space, since the columns of $\mathbf{B}^{-1}\mathbf{N}$ have m entries.

Step (vi) iterates through all the indexes s of columns \mathbf{a}_s in \mathbf{B} . For those indexes the value $\frac{x_s}{d_{sk}}$ needs to be checked for being negative, which can be done without actually computing the fraction. Moreover, comparing one such value with another can also be done without computing the value of the fraction. Finally, the column \mathbf{a}_s may have to be compared to another column vector for being lexicographically smaller. This can be done in space $\mathcal{O}(km)$ and hence the overall step can be executed in space $\mathcal{O}(km)$.

Finally, we note that updating the value for B requires only space logarithmic in m .

So overall, one optimization step of the revised Simplex Method requires not more than $\mathcal{O}(km^3)$ of space in addition to the space needed to store the original input. \square

It may be noted that the revised Simplex Method (and the Simplex Method as well) consists of two phases, from which we only have described the second. Since we suppose a starting solution \mathbf{x} , the description as it stands might not be very helpful in practice, since such a solution might not be given (and might not even exist). However, in our special circumstances we can explicitly give such a solution, as we shall see in the next subsection.

4.2 Deciding Entailment in Polynomial Space

With the revised Simplex Method at hand we are able to show that the linear programs (4) can be solved in polynomial space. The main obstacle is to transform (4) into the form of (5).

Let us consider the system (3). We can convert it into the form of (5) by adding *slack variables* y_i for each inequality in (1). More precisely, we can transform the inequalities (1) into the equations

$$\begin{aligned} \sum_{A \supseteq A_i} x_A - y_{i,1} &= s_i, \\ \sum_{A \supseteq A_i \cup B_i} x_A - c_i \sum_{A \supseteq A_i} x_A - y_{i,2} &= 0, \end{aligned}$$

such that $y_{i,j} \geq 0$ for all $i \in \{1, \dots, |I|\}, j \in \{1, 2\}$. We can do likewise for the inequalities in (2) by introducing y_1 and y_2 . Hence we obtain the system

$$\mathbf{Ax} - \mathbf{y} = \mathbf{b}, \quad \mathbf{x}, \mathbf{y} \geq \mathbf{0} \tag{6}$$

where $\mathbf{y} = ((y_{i,j})_{i \in I, j \in \{0,1\}}, y_1, y_2)$. The following observation is well known in the theory of linear programming.

Lemma 9 *If \mathbf{x} is a solution of (3), then $(\mathbf{x}, \mathbf{y}) \in \mathbb{Q}^{2^{|I|}+2|I|+2}$ is a solution of (6) where $\mathbf{y} = \mathbf{Ax} - \mathbf{b} \geq \mathbf{0}$. Conversely, if (\mathbf{x}, \mathbf{y}) is a solution of (6), then \mathbf{x} is a solution of (3).*

In particular, the linear programs (4) and the corresponding programs

$$\begin{aligned} & \min \left\{ \sum_{X \supseteq A} x_X \mid \mathbf{x} \in \mathcal{X} \right\}, \\ & \min \left\{ \sum_{X \supseteq A \cup B} x_X - c \sum_{X \supseteq A} x_X \mid \mathbf{x} \in \mathcal{X} \right\} \end{aligned} \tag{7}$$

with

$$\mathcal{X} = \{ \mathbf{x} \mid \mathbf{Ax} - \mathbf{y} = \mathbf{b}, \mathbf{x}, \mathbf{y} \geq \mathbf{0} \}$$

have the same values, respectively.

We are now ready to formulate the overall goal of our considerations.

Theorem 10 *The systems (7) can be solved in polynomial space.*

Corollary 11 *The linear programs (4) can be solved in polynomial space. Therefore, entailment of constrained implications can be decided in polynomial space.*

Proof (Theorem 10) Let $m = |I|$. Let us write the system (6) as

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}, \quad \tilde{\mathbf{x}} \geq \mathbf{0} \tag{8}$$

where $\tilde{\mathbf{A}} = (\mathbf{A}, -\mathbf{I}) \in \mathbb{Q}^{(2m+2) \times (2^m+2m+2)}$ and $\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{y}) \in \mathbb{Q}^{2^m+2m+2}$. Here, \mathbf{I} denotes the identity matrix of the corresponding size. Note that $\tilde{\mathbf{A}}$ has full rank, since it contains $-\mathbf{I}$ as submatrix.

We start by noting the following two facts, which we shall discuss in detail afterwards:

- i. The matrix $\tilde{\mathbf{A}}$ can be represented implicitly, i. e. the entries of $\tilde{\mathbf{A}}$ can directly be inferred from the input \mathcal{L} and r .
- ii. A solution for (8) can be given explicitly. This is due to the fact that we can find an explicit solution for (3), namely $x_M = 1$ and $x_A = 0$ for $A \subsetneq M$.

We start with point (i). For this let $i \in \{1, \dots, m+2\}$, $j \in \{1, \dots, 2^m+2m+2\}$ and $\tilde{\mathbf{A}} = (\tilde{a}_{st})_{s,t}$. Then we can distinguish the following cases:

1. If $j > 2^m$, then if $i = j - 2^m$, then $\tilde{a}_{ij} = 1$, otherwise $\tilde{a}_{ij} = 0$.
2. $i = 2m+1$ and $j \leq 2^m$, then $\tilde{a}_{ij} = 1$.
3. $i = 2m+2$ and $j \leq 2^m$, then $\tilde{a}_{ij} = -1$.

4. $i = 2k$ for some $k \in I$, $j \leq 2^m$. Let C be the set corresponding to the index j . Then $\tilde{a}_{ij} = 1$ if $C \supseteq A_k$ and $\tilde{a}_{ij} = 0$ otherwise.
5. $i = 2k + 1$ for some $k \in I$, $j \leq 2^m$. Again let C be the set corresponding to the index j . If $C \supseteq A_k \cup B_k$, then $C \supseteq A_k$ and $\tilde{a}_{ij} = 1 - c_k$. If $C \not\supseteq A_k \cup B_k$, but $C \supseteq A_k$, then $\tilde{a}_{ij} = -c_k$. Otherwise, $\tilde{a}_{ij} = 0$.

The last two cases can be seen from the inequalities (1). Hence, $\tilde{\mathbf{A}}$ can be inferred from \mathcal{L} and r and need not be stored explicitly.

For the point (ii) let \mathbf{x} as discussed there. Then let $\mathbf{y} = \mathbf{Ax} - \mathbf{b}$. Then $B = \{\mathbf{a}_{2^m}, \mathbf{a}_{2^m+1}, \dots, \mathbf{a}_{2^m+2m+1}\}$ is a set of linearly independent columns of \mathbf{A} that correspond to the solution (\mathbf{x}, \mathbf{y}) .

Now we can apply Theorem 8 to see that we only need $\mathcal{O}(km^3)$ additional space to solve (8), where k is the largest amount needed to store an entry of $\mathbf{A}, \mathbf{b}, \mathbf{c}$. This however reduces to the largest amount needed to store any value of s_i or c_i , which are part of the input. Since we have not used more than $\mathcal{O}(km^3)$ space so far, we can hence solve (8) in polynomial space in the size of the input and the claim is proven. \square

5 Conclusions and Further Research

Starting from the motivation to understand association rules as logical objects usable for reasoning, we have introduced the notion of constrained implications. Based on this definition we have developed semantics for constrained implications based on models and stated the corresponding entailment problem. We then investigated a reformulation of instances of the entailment problem as pairs of linear programs and were able to show that those linear programs can be solved in polynomial space.

The result that entailment for constrained implications can be decided in polynomial space does not seem that pleasing. Above all, it does not give reasonable constraints on the time needed to decide entailment, a fact that might be of practical interest.

A better complexity bound might hence be desirable. It is quite unclear whether one can expect the entailment problem to be an element of either \mathcal{NP} , $\text{co-}\mathcal{NP}$ or even \mathcal{P} . It might, however, be worth looking into complexity classes of the polynomial hierarchy whether the entailment problem is located there.

Another problem, which is closely connected to our original motivation but which has not been addressed in this work is the following: Since we have established a notion of entailment for constrained implications it might be worth searching for *minimal non-redundant* sets of constrained implications representing $\text{conImp}_{s,c}(\mathbb{K})$ for a formal context \mathbb{K} and $s, c \in [0, 1] \cap \mathbb{Q}$. Results in this direction might give more insight in the usefulness of regarding association rules as logical objects.

References

- [1] Bernhard Ganter and Rudolph Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin-Heidelberg, 1999.

- [2] Michael R. Garey and David S. Johnson. *Computers and Intractability*. W. H. Freeman and company, 1979.
- [3] C. Großmann and J. Terno. *Numerik der Optimierung*. Teubner Studienbücher. Teubner, 1997.
- [4] Martin Grötschel. Algorithmische Diskrete Mathematik II – Linear Optimierung. lecture notes, TU Berlin, 2004.
- [5] Marzena Kryszkiewicz. Concise representations of association rules. In David J. Hand, Niall M. Adams, and Richard J. Bolton, editors, *Pattern Detection and Discovery*, volume 2447 of *Lecture Notes in Computer Science*, pages 92–109. Springer, 2002.
- [6] M. Luxenburger. Partial implications. FB4-Preprint, TH Darmstadt, 1994.
- [7] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Intelligent structuring and reducing of association rules with formal concept analysis. In Franz Baader, Gerhard Brewka, and Thomas Eiter, editors, *KI/ÖGAI*, volume 2174 of *Lecture Notes in Computer Science*, pages 335–350. Springer, 2001.